PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

EX PARTE Shridhar Mukund et al.

Application for Patent

Filed December 2, 2003

Application No. 10/726,470

FOR:

NETWORKED PROCESSOR FOR A PIPELINED ARCHITECTURE

APPEAL BRIEF

CERTIFICATE OF EFILING

I hereby certify that this correspondence is being transmitted to the United States Patent and Trademark Office via electronic filing with EFS-Web on August 3, 2010.

Signed: _____ /Jose M. Nunez /_____
Jose M. Nunez

MARTINE PENILLA & GENCARELLA, LLP
Attorneys for Appellants

TABLE OF CONTENTS

Page No

I. REAL PARTY IN INTEREST 1

II. RELATED APPEALS AND INTERFERENCES 1

III. STATUS OF CLAIMS 1

IV. STATUS OF AMENDMENTS 1

V. SUMMARY OF CLAIMED SUBJECT MATTER 2

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL 5

VII. ARGUMENT 5

A. Rejection of claims 1, 2, 4-9, and 11-20 under 35 USC 103(a) over Narayan, Kregness, and Amagai. 6

1. Claims 1, 4-7, 9, and 11-13 6

2. Claims 14-20 10

3. Claims 2 and 8 13

B. Rejection of claim 21 under 35 U.S.C. 103(a) over Narayan, Kregness, Amagai, and Cheah 15

1. Claim 21 15

C. Conclusion 16

VIII. CLAIMS APPENDIX 17

IX. EVIDENCE APPENDIX 22

X. RELATED PROCEEDINGS APPENDIX 23

I. REAL PARTY IN INTEREST

The real party in interest is PMC-Sierra, Inc., which recently acquired Adaptec, Inc., the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

The Appellants are not aware of any related appeals or interferences.

It is noted that this is the third Appeal Brief filed for this Patent Application. Appellants filed previous Appeal Briefs on January 22, 2008, and April 27, 2009. In both instances the Examiner submitted new Non-Final Rejections and the Appeals did not reached the BPAI.

III. STATUS OF CLAIMS

Claims 1, 2, 4-9, and 11-21 are pending, with claims 1, 7, and 14 being independent. Claims 3 and 10 have been cancelled. Claims 1, 2, 4-9, and 11-21 have been rejected and are on appeal.

IV. STATUS OF AMENDMENTS

Appellants submitted an amendment on October 30, 2009, in response to a non-Final Office Action mailed on August 19, 2009. This amendment was the last entered amendment.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The subject invention is directed towards methods and apparatus for a processor configured to efficiently process incoming or outgoing packet data. The processor is arranged in a pipeline architecture, where one or more of the processors may be associated with a certain stage of the pipeline. The processor pipeline offloads previous processing performed by a central processing unit (CPU) of a host system, thereby freeing the CPU for other processing to improve system performance. The processor is configured to allow a single cycle access to a large address space.

Accordingly, a networking application processor as recited in **claim 1** includes an input socket configured to receive data *(see Figures 4 and 5, and page 9, lines 8-20)* associated with a processing stage of a data packet, each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet (*see Figure 3, page 8, lines 9-23, and page 11, first paragraph*) and a memory for storing instructions *(See Figures 4 and 5 and page 12 lines 16-21).* The processor includes circuitry configured to access data structures associated with the received data and with the corresponding network protocol, the circuitry configured to access data structures and enabling single clock cycle access of an operand from memory during consecutive clock cycles *(see Figures 6 and 7 and page 17, lines 15-25).* An arithmetic logic unit (ALU) *(Figures 6 and 8 and page 19, lines 5-10)* and circuitry for aligning operands to be processed by the ALU, the circuitry for aligning operands causing the operands to be aligned by a lowest significant bit, wherein the circuitry for aligning the operands supplies an extension to the operands to allow the ALU to process different size operands *(see Figure 6 and page 23, lines 10-21).*

Further, **claim 7** defines a processor that includes an input socket configured to receive data *(see Figures 4 and 5, and page 9, lines 8-20)* associated with a processing stage of a data packet, each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet (*see Figure 3, page 8, lines 9-23, and page 11, first paragraph*) and a memory for storing instructions *(see Figures 4 and 5 and page 12 lines 16-21).* The processor includes circuitry configured to access data structures associated with received data and with the corresponding network protocol, the circuitry configured to access data structures and enabling single clock cycle access of an operand from memory during consecutive clock cycles *(see Figures 6 and 7 and page 17, lines 15-25).* An arithmetic logic unit (ALU) *(Figures 6 and 8 and page 19, lines 5-10)* is also included. The ALU is configured to receive a first and a second operand where the second operand is specified from an internal register *(see Figure 6 and page 18, lines 9-21)* and the first operand being masked by a mask in the corresponding instruction to enable the ALU to process a non-masked segment of the first operand *(see Figure 6 and page 21, lines 14-25).*

Further yet, **claim 14** defines a processor capable of processing data associated with a processing stage of a pipeline of processors. The processor includes a data random access memory (RAM) configured to enable access to data structures *(see Figures 5 and 6 and page 12, lines 14-20).* Instruction fetch and decode circuitry configured to interpret instructions *(see Figures 5 and 6 and pages 13-14)* to be executed by an arithmetic logic unit (ALU) *(Figures 6 and 8 and page 19, lines 5-10)* is included. The instruction fetch and decode circuitry includes a read only memory (ROM) *(see Figures 5 and 6 and page 12, lines 14-20)*, the ROM configured to store code common to each processing stage

associated with a pipeline of processors. The instruction and fetch decode circuitry includes a code RAM *(see Figures 5 and 6 and page 12, lines 14-20)* configured to download code specific to the processing stage and wherein the code specific to the processing stage is enabled for single cycle access. The processor includes instruction decode circuitry *(see Figures 5 and 6 and pages 13-14)* configured to recognize operating instructions. Execute and write back circuitry *(see Figures 5 and 6 and page 12, lines 14-20 and page 13, lines 3-15)* configured to set up operands to be processed by the ALU is included. The execute and write back circuitry includes internal registers *(see Figure 6 and page 18, lines 9-21)* for defining a first and a second operand. The processor further includes an arithmetic logic unit *(Figures 6 and 8 and page 19, lines 5-10)* for processing the first and second operands and align function circuitry *(see Figures 6 and 8 and pages 22-23)* for aligning the first and the second operands to be processed by the ALU. The align function circuitry causes the first and the second operands to be aligned by a lowest significant bit, wherein the align function circuitry supplies an extension to the each of the operands to allow the ALU to transparently process different size operands *(see page 23 lines 3-20)*, where the execute and write back circuitry executes an instruction while the instruction fetch and decode circuitry fetches a next instruction of the instruction being executed, wherein the processor processes data associated with the processing stage of a data packet (*page 13, lines 7-22*), each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet (*see Figure 3, page 8, lines 9-23, and page 11, first paragraph*).

As referred to in this application, each of the pipelined processors include both hardware such as, input socket interface, star processor, output socket interface, hardware

accelerator, and relevant software to access the hardware. *(Figure 3 and related description on page 8, line 15 to page 11, line 5).* For the pipelined processors the output socket interface of a first processor is in communication with an input socket interface of a second processor, and so on *(see Figure 2).*

It should be appreciated that the above description represents only a summary of the present invention. A more in-depth discussion of the present invention is provided in the Detailed Description section of the application.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The following grounds of rejection are presented for review:

A. Whether claims 1, 2, 4-9, and 11-20 are patentable under 35 USC 103(a) over Narayan et al. (U.S. Patent No. 5,822,559) (hereinafter "Narayan"), in view of Kregness et al. (U.S. Patent No. 4,595,911) (hereinafter "Kregness"), and further in view of Amagai et al. (U.S. Patent No. 7,130,312) (hereinafter "Amagai"); and

B. Whether claim 21 is patentable under 35 U.S.C. 103(a) over Narayan, Kregness, Amagai, and further in view of Cheah (U.S. Patent No. 6,243,291) (hereinafter "Cheah").

VII. ARGUMENT

Appellants present the following arguments with respect to the rejected claims:

A. Rejection of claims 1, 2, 4-9, and 11-20 under 35 USC 103(a) over Narayan, Kregness, and Amagai.

1. Claims 1, 4-7, 9, and 11-13

i. The prior art does not disclose an input socket configured to receive data associated with a processing stage of a data packet, each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet

Claims 1 and 7 define an input socket configured to receive data associated with a processing stage of a data packet, where each processing stage of the data packet is associated with one network protocol from a plurality of network protocols in the data packet. The Office has asserted the following in reference to Fig. 2 of Narayan:

> "Any input to the processor is an input socket. This includes, but is not limited to: Connection from main memory to Prefetch Unit 202, Connection to main memory from Data Cache 224, clock signal, I/O requests, etc…. Additionally, Narayan discloses specific paths for communication; see Fig. 67 and col. 209, lines 1-9. Finally a processor inherently must receive data [including a program to execute] in order to perform useful calculations)" (page 2, last para. spanning to the beginning of the next page, emphasis added).

Appellants respectfully assert that the Examiner's rejection is ambiguous and that the Examiner has failed to consider the claim as a whole. "[A] patentable invention may lie in the discovery of the source of a problem even though the remedy may be obvious once the source of the problem is identified. This is part of the 'subject matter as a whole' which should always be considered in determining the obviousness of an invention under 35 U.S.C. § 103." *In re Sponnoble*, 405 F.2d 578, 585, 160 USPQ 237, 243 (CCPA 1969).

The Office has failed to analyze Appelants' claims as a whole, instead breaking down rejections into pieces that lack logic and do not show how the prior art teaches the featured claims. The Examiner seems to imply that any input to the processor suggests the claimed input socket. However, the claimed input socket is defined with certain characteristics, namely an input socket configured to receive data associated with **a**

processing stage of a data packet, where each processing stage of the data packet being **associated with one network protocol** from a plurality of network protocols in the data packet. The person skilled in the art will readily appreciate that a clock signal input to the processor is used to receive data associated with a data packet, because a clock signal input is only used to receive a clock signal. Even if Narayan were combined with Amagai (see below for more details), it would not be reasonable to deduce that a clock signal would be used to receive a data packet, as the processor would stop operating properly if the clock signal wasn't a continuous periodic sequence of high and low voltage signals.

Further, the Examiner further asserts in the rejection of this feature that Narayan teaches that [the input socket] is "configured to receive data (the data received by the processor including instructions and data associated with the instructions)" (page 3, first para., emphasis added). Appellants respectfully disagree. Data received by the processor that includes instructions and data associated with the instructions does not suggest data data associated with … a data packet. Instructions and data packets are different types of data because instructions dictate the operations performed by the processor, and data packets are related to information exchanged by processes. Thus, the Examiner's rejection is improper because the Examiner is referring to processor instructions and not data packets.

Further, the Examiner admits that "Narayan does not expressly disclose the data being associated with a processing stage of a data packet, each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet" (page 4, 4th para.). The Examiner asserts that Amagai teaches this feature in Figs. 4a-4c, and that "it would have been obvious … to have modified … Narayan …

[with] Amagai, in order to facilitate inter-computer communication. Appellants respectfully disagree. The Examiner is using improper hindsight to reconstruct the claims, and the combination suggested would not produce the claimed feature.

Amagai teaches that "Figs. 4A, 4B, and 4C [refer to] a data structure corresponding to Ethernet protocol" (page 7, col. 35-36, emphasis added). The Examiner is combining a processor, which is a tangible item, with a data structure, which is an intangible item. This appears impossible as combining tangible and intangible items does not seem feasible. Further, merely reciting a data structure would not suggest to a person skilled in the art to add an input socket to a processor, where the input socket is dedicated to just one stage of processing of a data packet. The Examiner has not articulated reasoning with rational underpinning to support the conclusion of obviousness. The Supreme Court in KSR noted that the analysis supporting a rejection under 35 U.S.C. 103 should be made explicit, *In re KSR International Co. v. Teleflex Inc.*, 550 USPQ2d 1385 (2007). The Court in KSR quoted *In re Kahn*, which stated that ''[R]ejections on obviousness cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.''

For all these reasons, the prior art does not teach an input socket configured to receive data associated with a processing stage of a data packet, where each processing stage of the data packet is associated with one network protocol from a plurality of network protocols in the data packet, as claimed by Appellants.

ii. Narayan does not disclose circuitry configured to access data structures associated with the received data and with the corresponding network protocol

Claims 1 and 7 define circuitry configured to access data structures associated with the received data and with the corresponding network protocol (emphasis added). The Examiner has failed again to consider the claim as a whole and proceeded to break the claimed feature in unrelated atomic elements. The Examiner as asserted that Narayan teaches the following:

> "circuitry (such as Instruction Alignment Unit; see Fig. 2) configure to access data structures (such as instructions or data associated with a program) associated with the received data (instruction and program is associated with data received)" (page 3, first para., emphasis added).

In addition, the Examiner has again referred to the data structures in Figures 4A-4C of Agamai to teach that the data is associated with the corresponding network protocol. Appellants respectfully disagree.

First, if the circuitry that teaches the claimed circuitry is an Instruction Alignment Unit (IAU), no person skilled in the art would assert that an IAU accesses data structures, since an IAU only accesses one instruction at a time to align the instruction. The person skilled in the art would appreciate that a data structure can include more than one word. Thus, and IAU does not access data structures because the IAU can only access one instruction at a time.

Second, the combination of Narayan and Agamai would not produce the claimed feature, for the same reasons described above in reference to argument (i). Combining Ethernet related data structures with an IAU would not produced circuitry configured to access data structure associate with … the corresponding network protocol. There is no

suggestion in either of the references cited that indicates hardware specific for a network protocol. The Examiner's assertion is improper hindsight, as previously discussed.

2. Claims 14-20

i. Narayan does not disclose a processor capable of processing data associated with a processing stage of a pipeline of processors

Claim 14 defines a processor capable of processing data associated with a processing stage of a pipeline of processors. The Office has asserted that Narayan teaches the pipeline of processors because "each pipeline stage does processing and therefore any pipelined processor contains a 'pipeline of processors.'" Additionally, the Examiner has identified the following excerpt:

> "As used herein, a 'clock cycle' is an interval of time accorded to various stages of an instruction processing pipeline within the microprocessor to complete their various functions. Storage devices (e.g. registers and arrays) capture their values according to a clock signal defining the clock cycle." (col. 1, lines 15-20 - emphasis added).

Appellants respectfully disagree. Narayan teaches an instruction processing pipeline within the microprocessor, which nowhere suggests a pipeline of processors. Suggesting that any pipelined processor contains a pipeline of processors is simply illogical and untrue. The Office is offering a twisted play on words, which any reasonable person skilled in the art would consider inappropriate. Not everything in a system that performs a function is a processor. For example, memory performs a function of storing data, yet a person skilled in the art would never call a memory a processor. The Office has failed to show how Narayan teaches a processor capable of processing a data packet associated with a processing stage of a pipeline of processors, and the Office's rejection is improper.

Further, in the Response to Arguments section of the Office Action dated January 7, 2009, the Examiner has asserted the following:

> 'Examiner disagrees. The definition of a processor is "A part of a computer, such as the central processing unit, that performs calculations or other manipulations of data" ... A processing stage is by definition, a processor, not because of a "twisted play on words" as applicant alleges' (page 13, third paragraph - emphasis added).

Appellants respectfully disagree. Using the same reference as used by the Examiner, Dictionary.com defines stage as "*a single step or degree in a process; a particular phase, period, position, etc., in a process, development, or series.*" If a processor is "a part of a computer" (as cited by the Examiner) then, which part of the computer would the stage be? "A part of a computer" denotes that a processor is something tangible. However, a stage is not tangible because it refers to a "step or degree in a process," and a process is something intangible. It is not possible to touch or hold a "step or degree in a process." Thus, if a stage is not tangible, a stage cannot be a part of a computer and cannot be a processor.

Further yet, in the Response to Arguments section of the Office Action dated August 19, 2009, the Examiner has asserted the following:

> "Examiner respectfully disagrees. … Examiner did not mean the intangible step in an execution, but rather the computer science definition of the hardware associated with a step" (page 12, 2nd para., emphasis added)

Appellants are not aware of any computer science definition of the hardware associated with a step. If there is such a thing, Appellants respectfully request that the Examiner provides such definition of the hardware associate with a step. Since the Examiner has not provided a reference to indicate what is the "hardware associated with a

step", it seems that the Examiner has provided an Official Notice. Appellants respectfully traverse. Note excerpt from MPEP below.

"If the applicant traverses such an [Official Notice] assertion the examiner should cite a reference in support of his or her position." See MPEP 2144.03.

Appellants do not believe that such thing exists. A step is something in a process, and computer systems execute processes using a variety of hardware components. There is no standard definition of the hardware associated with a stage.

ii. Narayan does not disclose that the instruction fetch and decode circuitry includes a read only memory (ROM) configured to store code common to each processing stage associated with a pipeline of processors

Further, claim 14 defines that the instruction fetch and decode circuitry includes a read only memory (ROM), the ROM being configured to store code common to each processing stage associated with a pipeline of processors. The Examiner has asserted that the ROM is anticipated by Narayan in "such as portion of the MROM … system comprising the microcode sequences" (page 5, 2nd para.). However, the Examiner has not indicated how the prior art teaches that the ROM is configured to store code common to each processing stage associated with a pipeline of processors. Appellants respectfully assert that the prior art does not teach that the ROM is configured to store code common to each processing stage associated with a pipeline of processors. There is no teaching in Narayan that the MROM is configured to store code common to each processing stage associated with a pipeline of processors. Thus, the prior art does not teach the aforementioned feature.

iii. The prior art does not disclose that the instruction fetch and decode circuitry includes a code RAM configured to download code specific to the processing stage

Claim 14 defines that the instruction fetch and decode circuitry includes a code RAM configured to download code specific to the processing stage. The Examiner has asserted that the code RAM is taught by Narayan in "[t]he section of MROM unit 209 storing MROM instructions" (page 6, 1st para.) Appellants respectfully disagree. The Examiner has asserted that the ROM is also taught by a section of the MROM and that the MROM stands for "Microcode ROM" (page 5, last para.). However, if the MROM is microcode ROM and teaches a ROM (Read Only Memory), then the MROM cannot suggest a code RAM, because ROM and RAM are different types of memory where the RAM can be written to during normal operation, while the ROM is read only memory that cannot be overwritten.

3. <u>Claims 2 and 8</u>

i. The prior art does not disclose the feature wherein the single cycle access enables the data to be addressed and operated on in a single clock cycle without being placed into a register

Dependent claims 2 and 8 define the feature where the single cycle access enables the data to be addressed and operated on in a single clock cycle without being placed into a register (emphasis added). The Office has asserted that this feature is taught by Narayan as follows:

> "… see Fig. 20 regarding fetching and scanning in the same clock cycle and col. 64, lines 23-25 regarding fetching and muxing alignment).
>
> Note the definition of the word "operate" according to the American Heritage® Dictionary of the English Language, Fourth Edition is to 'perform a function; work'. Under this definition,

> a read (fetch) from memory is reasonable considered to be an operation." (page 8, para. 1-1).

Appellants respectfully disagree. The Examiner is trying to redefine the basic principles of computer science. No person skilled in the art would consider that fetching data suggests that data is being operated upon. Appellants understand that claims should be given the broadest reasonable interpretation consistent with the specification. However, "[t]he broadest-construction rubric ... does not give the PTO an unfettered license to interpret claims to embrace anything remotely related to the claimed invention. Rather, claims should always be read in light of the specification and teachings in the underlying patent," *In re Suitco Surface*, (Fed. Cir. Case No. 2009-1418, April 14, 2010). Further, the Supreme Court in *KSR* noted that the analysis supporting a rejection under 35 U.S.C. 103 should be made explicit, *In re KSR International Co. v. Teleflex Inc. (KSR),* 550 U.S. _, 82 USPQ2d 1385 (2007). The Court in KSR quoted *In re Kahn*, which stated that ''[R]ejections on obviousness cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness" (emphasis added).

The Examiner must interpret the claims according to the Specification and that which is well known in the art. Neither the Specification nor a person skilled in the art would interpret that reading data is an operation on the data. The data is operated upon when the data is transformed in some way or used in a computer process. A mere read does not operate on the data as the data is not part of any operation or computer process. Thus, the Examiner's rejection is improper.

B. Rejection of claim 21 under 35 U.S.C. 103(a) over Narayan, Kregness, Amagai, and Cheah

1. Claim 21

i. The prior art does not disclose that the execute and write back operation is executed simultaneously with the instruction fetch and decode operation for a next instruction following the execute and write back operation

Claim 21 specifies that the execute and write back operation is executed simultaneously with the instruction fetch and decode operation for a next instruction following the execute and write back operation. The Examiner has asserted the following:

> "Cheah teaches an instruction fetch and decode operation (Input and Address decoding: see fig. 4 and fig. 5), and an execute and writeback operation (Word address switching and Output), wherein the execute and write back operation is executed simultaneously with the fetch and decode operation for a next instruction following the execute and write back operation (see fig. 5)" (page 12, 4th para., emphasis added).

Appelants respectfully disagree. The Examiner has improperly interpreted the teachings of Cheah, because Figures 4 and 5 only refer to reading from memory in two stages. Cheah teaches the following:

> "As noted above, to reduce current drain and improve performance of the memory device 400, only 32 bits of a 64-bit page are sensed at a given time. A two-stage pipeline process is used to read the entire 64-bit page within the same 190 ns used by the prior art device, illustrated in FIG. 3. In FIG. 5, while the first two words (32 bits) are being output from the memory device 400, the second 32 bits are being sensed" (col. 6, lines 37-44, emphasis added).

Cheah teaches to read 32 bits at a time of a 64-bit page, which is performed as a two-stage pipeline. Figures 4 and 5 of Cheah do not teach that write back operation is executed simultaneously with the instruction fetch and decode operation for a next instruction following the execute and write back operation, because Figures 4 and 5 only teach about reading one 64-bit word of memory. There is not teaching regarding executing and writing back while the next instruction is being fetched.

C. Conclusion

In view of the foregoing reasons, the Appellants submit that each of the claims 1, 2, 4-9, and 11-21 are patentable. Therefore, the Appellants respectfully request that the Board of Patent Appeals and Interferences reverse the Examiner's rejections of the claims on appeal.

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, LLP

/Jose M. Nunez/

Jose M. Nunez
Reg. No. 59,979

710 Lakeway Drive, Suite 200
Sunnyvale, CA 94085
Telephone: (408) 749-6900
Facsimile: (408) 749-6901
Customer Number 25920

VIII. CLAIMS APPENDIX

1. A networking application processor, comprising:

an input socket configured to receive data associated with a processing stage of a data packet, each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet;

a memory for storing instructions;

circuitry configured to access data structures associated with the received data and with the corresponding network protocol, the circuitry configured to access data structures and enabling single clock cycle access of an operand from memory during consecutive clock cycles ;

an arithmetic logic unit (ALU); and

circuitry for aligning operands to be processed by the ALU, the circuitry for aligning operands causing the operands to be aligned by a lowest significant bit, wherein the circuitry for aligning the operands supplies an extension to the operands to allow the ALU to process different size operands.

2. The networking application processor of claim 1, wherein the instructions have a width of 96 bits, and wherein the single cycle access enables the data to be addressed and operated on in a single clock cycle without being placed into a register.

4. The networking application processor of claim 1, further including:

an output socket for transmitting processed data; and

a 64 bit bus connecting the input socket and the output socket.

5. The networking application processor of claim 1, wherein the extension to the operand fills each higher bit with a value.

6. The networking application processor of claim 1, wherein the operand is selected from the group consisting of a source operand, a destination operand, an immediate operand, and an internal register operand.

7. A processor, comprising:

an input socket configured to receive data associated with a processing stage of a data packet, each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet;

a memory for storing instructions;

circuitry configured to access data structures associated with received data and with the corresponding network protocol , the circuitry configured to access data structures and enabling single clock cycle access of an operand from memory during consecutive clock cycles ; and

an arithmetic logic unit (ALU), the ALU configured to receive a first and a second operand; the second operand being specified from an internal register, the first operand being masked by a mask in the corresponding instruction to enable the ALU to process a non-masked segment of the first operand.

8. The processor of claim 7, wherein the instructions have a width of 96 bits, and wherein the single cycle access enables the data to be addressed and operated on in a single clock cycle without being placed into a register.

9. The processor of claim 7, wherein each of the instructions include a loadback feature enables random accesses to one of a source indirect register or a destination indirect register through indirect addressing.

11. The processor of claim 7, wherein the first and the second operands are associated with a size selected from the group consisting of 8 bit operands, 16 bit operands, and 32 bit operands.

12. The processor of claim 7, wherein the first operand is selected from the group consisting of a source operand, a destination operand, an immediate operand, and an internal register operand.

13. The method of claim 7, wherein the memory location is a static random access memory (SRAM).

14. A processor capable of processing data associated with a processing stage of a pipeline of processors, the processor comprising:

a data random access memory (RAM) configured to enable access to data structures;

instruction fetch and decode circuitry configured to interpret instructions to be executed by an arithmetic logic unit (ALU) , the instruction fetch and decode circuitry including,

a read only memory (ROM), the ROM configured to store code common to each processing stage associated with a pipeline of processors;

a code RAM, the code RAM configured to download code specific to the processing stage and wherein the code specific to the processing stage is enabled for single clock cycle access; and

instruction decode circuitry configured to recognize operating instructions;

execute and write back circuitry configured to set up operands to be processed by the ALU, the execute and write back circuitry including,

internal registers for defining a first and a second operand;

an arithmetic logic unit for processing the first and second operands; and

align function circuitry for aligning the first and the second operands to be processed by the ALU, the align function circuitry the circuitry causing the first and the second operands to be aligned by a lowest significant bit, wherein the align function circuitry supplies an extension to the each of the operands to allow the ALU to transparently process different size operands, wherein the execute and write back circuitry executes an instruction while the instruction fetch and decode circuitry fetches a next instruction of the instruction being executed, wherein the processor processes data associated with the processing stage of a data packet, each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet.

15. The processor of claim 14, wherein the extension to each of the operands fills each higher bit with a value.

16. The processor of claim 14, wherein the different size operands have a width selected from the group consisting of 8 bits, 16 bits, and 32 bits.

17. The processor of claim 14, wherein the operating instructions wherein the operating instructions are formatted as 96 bit instructions, each of the 96 bit instructions including a single return bit.

18. The processor of claim 14, wherein the processor is configured as a two stage pipeline for pipelining an instruction fetch and decode operation and an execute and write back operation.

19. The processor of claim 14, wherein the operating instructions include microcode configured to predict a likely direction for a branch instruction.

20. The processor of claim 19, wherein no operation (NOP's) instructions are included, the NOP's configured block an invalidated pre-fetched instruction.

21. The networking application processor of claim 1, wherein the single clock cycle access includes simultaneous operations of:

an instruction fetch and decode operation, and

an execute and write back operation, wherein the execute and write back operation is executed simultaneously with the instruction fetch and decode operation for a next instruction following the execute and write back operation.

IX. EVIDENCE APPENDIX

There is currently no evidence entered and relied upon in this Appeal.

X. RELATED PROCEEDINGS APPENDIX

There are currently no decisions rendered by a court or the Board in any proceeding identified in the Related Appeals and Interferences section.